REMARKS / ARGUMENTS

Applicants have reviewed the Office Action dated March 8, 2007 (hereinafter "Office Action"), in which the Examiner: 1) rejected Claims 1-4, 7-8, 11-12, and 15-16 under 35 U.S.C. § 102(a) as allegedly anticipated by T. Okada (Patent No. U.S. 7,114,151 B2)(hereinafter *Okada*) and 2) rejected Claims 5-6, 9-10, 13-14 and 17-18 under 35 U.S.C. § 103(a) as allegedly obvious over *Okada* in view of Imai et al. (Patent No. U.S. 6,367,067 B1)(hereinafter *Imai*).

Claims 1-18 as originally filed remain pending in this application. Based upon the arguments presented below, Applicants believe all claims to be in condition for allowance.

Claim Rejections Under 35 U.S.C. § 102(a)

In the Office Action, the Examiner rejected Claims 1-4, 7-8, 11-12, and 15-16 as allegedly anticipated by *Okada*. Specifically, the Examiner stated:

> "As to claim 1, Okada discloses a code generating system, comprising: a compiler that receives source code (Fig. 9, steps of 'source program', SI — compiler; Col. 6, Lines 59-63) and generates an object file comprising object code (Fig. 9, step of 'assembly language code optimizer') and intermediate code (Fig. 10, step of S11 — generate intermediate code); a code optimizer coupled to the compiler (Fig. 9, step of S2 — code optimizer; Fig. 10, steps of S12 through S18 — code optimization) and a linker that receives the object file comprising object code and intermediate code (Fig. 9, step of S4 — linker; Col. 7, Lines 5-9) and provides the intermediate code to the code optimizer (Fig. 10, step of S11 — generate intermediate code; Col. 7, Lines 36-37; Fig. 12 - a flow of operations made in generation of the intermediate code; Col. 7, Lines 12-20).
>
> "As to claim 8, Okada discloses a method to optimize a program consisting of a plurality of source files (Fig. 9, steps of `source program'), the method comprising: producing intermediate code associated with one or more of the plurality of source files (Fig. 10, step of S11 – generate intermediate code); producing object code associated with one or more of the plurality of source files (Fig. 9, step of 'assembly language code optimizer'); merging the intermediate code and the object code associated with each source file into an object file comprising object code plus intermediate code (Fig. 10, step of S11 – generate intermediate code, steps of S12 through S18 – code optimization); and optimizing the program by providing the intermediate code in the object file to a code optimizer (Fig. 9, step of S2 – code optimizer; Fig. 10, step of S11 – generate intermediate code, steps of S12 through S18 – code optimization; Col. 7, Lines 12-20).

> "As to claim 12, Okada discloses a storage medium containing instructions that are executed by a processor and comprising: instructions that produce intermediate code from one or more source files (Fig. 10, step of S11 – generate intermediate code); instructions that produce object code from one or more source files (Fig. 9, step of `assembly language code optimizer'); instructions that merge the intermediate code and the object code associated with one of the source files into a single intermediate plus object code file (Fig. 10, step of S11 – generate intermediate code, steps of S12 through S18 – code optimization); and instructions that provide the intermediate code contained in the single intermediate plus object code file to a code optimizer (Fig. 9, step of S2 – code optimizer; Fig. 10, step of S11 – generate intermediate code, steps of S12 through S18 – code optimization; Col. 7, Lines 12-20).
>
> "As to claim 16, Okada discloses a computer system, comprising: a processor, memory coupled to the processor; a code generating system stored in the memory and executable on the processor and that produces intermediate code (Fig. 10, step of S11 – generate intermediate code) and object code (Fig. 10, S18 – generate assembly language code) that is stored into a single intermediate plus object code file and provided to a code optimizer (Fig. 9, step of S2 – code optimizer; Fig. 10, step of S11 – generate intermediate code, steps of S12 through S18 – code optimization; Col. 7, Lines 12-20)." (Office Action, pages 2-4)

Applicants respectfully traverse this rejection. Anticipation under Section 102 can be found only if a single reference shows exactly what is claimed. See *Titanium Metals Corp. v. Banner*, 227 U.S.P.Q. 773 (Fed. Cir. 1985). For a reference to anticipate under Section 102, ever element of the claimed invention must be identically shown in a single reference. See In re Bond, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990). That is, the prior art reference must show the *identical invention "in as complete detail as contained in the...claim"* to support a prima facie case of anticipation. *Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q. 2d 1913, 1920 (Fed. Cir. 1989)(emphasis added). Thus, for anticipation, the cited reference must not only disclose all of the recited features but must also disclose the *part-to-part relationships* between these features. See *Lindermann Maschienfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 486 (Fed. Cir. 1984). Accordingly, the Applicants need only point to a single element or claimed relationship not found in the cited reference to demonstrate that the cited reference fails to anticipate the claimed subject matter. A *strict correspondence* between the claimed language and the cited reference must be established for a valid anticipation rejection.

Moreover, the Applicants submit that, during the patent examination, the pending claims must be given an interpretation that is reasonable and consistent with the specification. See *In re Prate*, 162 U.S.P.Q. 541, 550-51 (C.C.P.A. 1969); *In re Morris*, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 19970; see also M.P.E.P. §2111 (describing the standards for claim interpretation during prosecution). Indeed, the *specification* is "the primary basis for construing the claims." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005). It is usually dispositive. *See id.* Interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *See In re Cortright*, 49 U.S.P.Q.2d 1464, 1468 (Fed. Cir. 1999); *see also* M.P.E.P. §2111. That is, recitations of a claim must be read as they would be interpreted by those of ordinary skill in the art. See *Rexnord Corp. v. Laliram Corp.*, 60 U.S.P.Q.2d 1851, 1854 (Fed. Cir. 2001); *see also* M.P.E.P.§2111.01. In summary, an Examiner, during prosecution, must interpret a claim recitation as one of ordinary skill in the art would reasonably interpret the claim in view of the specification. *See In re American Academy of Science Tech Center*, 70 U.S.P.Q.,2d 1827 (Fed. Cir. 2004).

Independent Claim 1 recites, "[a] code generating system, comprising:

a compiler that receives source code and generates an object file comprising object code and intermediate code;

a code optimizer coupled to the compiler; and

a linker that receives the object file comprising object code and intermediate code and provides the intermediate code to the code optimizer."

Numerous claimed limitations and element relationships are missing in *Okada*. Selecting just one: *Okada* does not disclose – or suggest – at least "...a linker that receives the object file comprising object code and intermediate code and provides the intermediate code to the code optimizer." From a careful review of *Okada*, it appears the output from the *Okada* linker **S4** not only fails to provide intermediate code to the code optimizer **S2**, but the *Okada* linker **S4** fails to provide anything at all to code optimizer **S2**. (*Okada*, Figure 9) Moreover, according to *Okada*: "FIG. 10 shows a flow of operations made **in the assembly language code optimizer** shown in FIG. 9." (Column 5, lines 47-48;

emphasis added) *Okada* then continues with Figs. 11-32, and the associated description in the specification, to explain in more detail the specific method of FIG. 10. Thus everything of Okada identified by the Examiner, regardless of whether Applicants agree with the various applications and interpretations of the Okada reference, is performed within the code optimizer. As such, *Okada* cannot anticipate independent Claim 1 under Section 102.

Independent Claim 8 recites, "[a] method to optimize a program consisting of a plurality of source files, the method comprising:

producing intermediate code associated with one or more of the plurality of source files;

producing object code associated with one or more of the plurality of source files;

merging the intermediate code and the object code associated with each source file into an object file comprising object code plus intermediate code; and

optimizing the program by providing the intermediate code in the object file to a code optimizer."

While the Examiner's interpretation of *Okada* for this claim appears to run counter to that set forth in rejecting Applicants' independent Claim 1, numerous claimed limitations and element relationships of independent Claim 8 are also missing in *Okada*. Selecting just one: *Okada* does not disclose – or suggest – at least "...optimizing the program by providing the intermediate code in the object file to a code optimizer". Applicants' claimed invention "merg[es] the intermediate code and the object code associated with each source file into an object file comprising object code plus intermediate code" which "intermediate code in the object file" is provided "to a code optimizer". Assuming that the Examiner's interpretation of the *Okada* reference is used, with which interpretation the Applicants most respectfully disagree, the *Okada* method at a minimum stills fails to show "providing the intermediate code in the object file to a code optimizer" (emphasis added). As discussed above, according to *Okada*: "FIG. 10 shows a flow of operations made **in the assembly language code optimizer** shown in FIG. 9." (Column 5, lines 47-48; emphasis added). FIG 10 of *Okada* indicates that optimizer **S2** generates what *Okada*

refers to as intermediate code at step **S11.** This presents a problem: if optimizer **S2** generates "intermediate code", then the intermediate code cannot then be provided **to** optimizer **S2**; it is already in optimizer **S2.** This problem in turn presents other logical difficulties in applying *Okada* to independent Claim 8. For example, for "the intermediate code in the object file" to be provided "to a code optimizer" – as Applicants' have claimed – this means Applicants' claimed merging function is also not completed by the code optimizer **S2**. These claimed limitations and relationships are in sharp contrast to *Okada's* teachings; thus, *Okada* cannot anticipate independent Claim 8.

Independent Claim 12 recites, "[a] storage medium containing instructions that are executed by a processor and comprising:

instructions that produce intermediate code from one or more source files;

instructions that produce object code from one or more source files;

instructions that merge the intermediate code and the object code associated with one of the source files into a single intermediate plus object code file; and

instructions that provide the intermediate code contained in the single intermediate plus object code file to a code optimizer.

While the Examiner's interpretation of *Okada* for this claim appears to run counter to that set forth in connection with Applicants' independent Claim 1, numerous claimed limitations and element relationships of independent Claim 12 are also missing in *Okada*. Selecting just one: *Okada* does not disclose – or suggest – at least "...instructions that provide the intermediate code contained in the single intermediate plus object code file to a code optimizer". Applicants' claimed invention comprises "instructions that merge the intermediate code and the object code associated with one of the source files into a single intermediate plus object code file" which "intermediate code contained in the single intermediate plus object code file" is provided "to a code optimizer". Assuming that the Examiner's interpretation of the *Okada* reference is used, with which interpretation the Applicants most respectfully disagree, the *Okada* reference stills fails to show "instructions that provide the intermediate code contained in the single intermediate plus object code file to a code optimizer" (emphasis added). As discussed above, according

to *Okada*: "FIG. 10 shows a flow of operations made **in the assembly language code optimizer** shown in FIG. 9." (Column 5, lines 47-48; emphasis added). FIG 10 of *Okada* indicates that optimizer **S2** generates what *Okada* refers to as intermediate code at step **S11.** This again presents a problem: if optimizer **S2** generates "intermediate code", then the intermediate code – regardless of what kind of file in which is it contained - cannot then be provided **to** optimizer **S2**; it is already in optimizer **S2.** This problem in turn presents other logical difficulties in applying *Okada* to independent Claim 12. For example, for "the intermediate code contained in the single intermediate plus object code file" to be provided "to a code optimizer" – as Applicants' have claimed – this means Applicants' claimed instructions that merge are also not part of code optimizer **S2** itself. These claimed limitations and relationships are in sharp contrast to *Okada's* teachings. As such, *Okada* cannot anticipate independent Claim 12.

Claim 16 recites, "[a] computer system, comprising:

a processor,

memory coupled to the processor;

a code generating system stored in the memory and executable on the processor and that produces intermediate code and object code that is stored into a single intermediate plus object code file and provided to a code optimizer.

While the Examiner's interpretation of *Okada* for this claim appears to run counter to that set forth in connection with Applicants' independent Claim 1, numerous claimed limitations and element relationships of independent Claim 12 are still missing in *Okada*. Selecting just one, and assuming that the Examiner's interpretation of the *Okada* reference is used, with which interpretation the Applicants most respectfully disagree, the *Okada* reference still fails to disclose – or suggest – at least "…a code generating system …that produces intermediate code and object code that is stored into a single intermediate plus object code file and provided to a code optimizer". (emphasis added). As discussed above, according to *Okada*: "FIG. 10 shows a flow of operations made **in the assembly language code optimizer** shown in FIG. 9." (Column 5, lines 47-48; emphasis added). FIG 10 of *Okada* indicates that optimizer **S2** generates what *Okada*

refers to as intermediate code at step **S11.** This again presents a problem: if optimizer **S2** generates "intermediate code", then the intermediate code – regardless of what kind of file in which is it contained - cannot then be provided **to** optimizer **S2**; it is already in optimizer **S2.** These claimed limitations and relationships are in sharp contrast to *Okada's* teachings. As such, *Okada* cannot anticipate independent Claim 16.

Thus, *Okada* cannot anticipate any of independent Claims 1, 8, 12, and 16 under Section 102. Accordingly, Applicants respectfully request withdrawal of the rejection under Section 102 and allowance of independent Claims 1, 8, 12 and 16, as well as all claims depending therefrom.

Claim Rejections Under 35 U.S.C. §103(a)

In the Office Action, the Examiner rejected claims 5-6, 9-10, 13-14 and 17-18 under 35 U.S.C. § 103(a) as allegedly obvious over *Okada* in view of *Imai*. Applicants respectfully traverse these rejections.

The burden of establishing a prima facie case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (B.P.A.I. 1979). In establishing a *prima facie* case for obviousness, it is often necessary "to look to interrelated teachings of multiple patents, the effects of demands known to the design community or present in the market place; and the background knowledge possessed by a person having ordinary skill in the art." *KSR Int'l Co. v. Teleflex, Inc.* No. 04-1350, slip op. at 14 (U.S. April 30, 2007). Indeed, "the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or nonobviousness of the subject matter is determined." *Id.* at 2 (quoting *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966)). This analysis should be made explicit. *Id.* at 14 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006))("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness").

Additionally, a claim having several elements is *not* proved obvious merely by demonstrating that each of its elements was known in the prior art. *Id.* As such, the obviousness inquiry does not hinge on demonstrating that elements were known in the art. Rather, the obviousness inquiry focuses on whether the claimed subject matter would have been obvious to persons having ordinary skill in the art in view of the demands and practices of the design community at the time of filing of the application. *See id.*

Moreover, the Applicants submit that, during the patent examination, the pending claims must be given an interpretation that is reasonable and consistent with the specification. See *In re Prate*, 162 U.S.P.Q. 541, 550-51 (C.C.P.A. 1969); *In re Morris*, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 19970; see also M.P.E.P. §2111 (describing the standards for claim interpretation during prosecution). Indeed, the *specification* is "the primary basis for construing the claims." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005). It is usually dispositive. *See id.* Interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *See In re Cortright*, 49 U.S.P.Q.2d 1464, 1468 (Fed. Cir. 1999); *see also* M.P.E.P. §2111. That is, recitations of a claim must be read as they would be interpreted by those of ordinary skill in the art. See *Rexnord Corp. v. Laliram Corp.*, 60 U.S.P.Q.2d 1851, 1854 (Fed. Cir. 2001); *see also* M.P.E.P.§2111.01. In summary, an Examiner, during prosecution, must interpret a claim recitation as one of ordinary skill in the art would reasonably interpret the claim in view of the specification. *See In re American Academy of Science Tech Center*, 70 U.S.P.Q.,2d 1827 (Fed. Cir. 2004).

Specifically, the *Imai* reference fails to remedy any of the above-noted deficiencies of the *Okada* reference with which it is combined. As a result, neither *Okada* nor *Imai*, taken together or separately, render Claims 5-6, 9-10, 13-14 and 17-18 obvious. Accordingly, Applicants respectfully request withdrawal of the rejection under Section 103 and allowance of all claims.

CONCLUSION

Applicants respectfully submit that for at least the reasons presented herein, all claims are in condition for allowance. In the course of the foregoing discussions, Applicants may have at times referred to claim limitations in shorthand fashion, or may have focused on a particular claim element. This discussion should not be interpreted to mean that the other limitations can be ignored or dismissed. The claims must be viewed as a whole, and each limitation of the claims must be considered when determining the patentability of the claims. Moreover, it should be understood that there may be other distinctions between the claims and the cited art which have yet to be raised, but which may be raised in the future.

Applicants respectfully request reconsideration and that a timely Notice of Allowance be issued in this case. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to the Texas Instruments, Inc. Deposit Account No. 20-0668.

Respectfully submitted,

/L.Joy Griebenow/
L.Joy Griebenow
Reg. No. 33,704
CONLEY ROSE, P.C.
970-232-3093 (Phone)
Attorney for Applicants

Correspondence Address:
Texas Instruments Incorporated
Robert Marshall
Customer Number 23494
P. O. Box 655474, MS 3999
Dallas, TX 75265
Telephone: (972) 917-5290